

IN THE CLAIMS-

~~Claims 1-2 (canceled)~~

---

Claim 3 (previously amended): The medium recited in claim 106 wherein the code comprises an advertising tag, the network server comprises an advertising server and the information object comprises a web advertisement.

41  
Claim 4 (previously added): The medium recited in claim 3 wherein said one file comprises an Ad Descriptor file or at least one advertising file specified in the Ad Descriptor file, the advertising file being either a media file or a player file.

Claim 5 (previously added): The media recited in claim 4 wherein the advertising code comprises an advertising tag, which, when executed by the computer, causes the computer to dynamically write a plurality of predefined applet tags that collectively implement a script into the first web page, wherein the script, when subsequently executed by the computer, causes the computer to download an agent from a predefined distribution server into memory in the computer and thereafter instantiate and execute the agent.

Claim 6 (previously added): The media recited in claim 5 wherein the user-initiated event is an affirmative action taken by the user, through a web browser, to navigate from the first web page to the next successive web page, wherein the action comprises a mouse click, a key depression or a user-invoked state change in a stored history of web pages previously visited by the user.

Claim 7 (previously added): The media recited in claim 6 wherein the computer executes the web browser which, in turn, executes the first web page, comprising the tag, and subsequently the agent.

Claim 8 (previously added): The media recited in claim 7 wherein the agent overrides default life cycle methods defined in the web browser with corresponding substitute methods such that the agent persistently remains in browser storage as the browser transitions across successive web pages and different web sites.

*b1  
Revised*  
Claim 9 (previously added): The media recited in claim 8 wherein the agent comprises:

- a Transition Sensor applet; and
- an Ad Controller applet;

wherein the Transition Sensor applet instantiates and starts execution of the Ad Controller applet, monitors a user click-stream so as to detect the user-initiated event, and:

- instructs the Ad Controller applet to download the Ad Descriptor file for the web advertisement from the advertising server into the browser storage on the computer; and

- in response to an occurrence of the event, instructs the Ad Controller applet to cease any download of a further advertisement file specified in the Ad Descriptor file, to the extent any downloading of said further advertisement file is then occurring, and to initiate processing, through the browser, of files for an advertisement that has been previously downloaded and is currently ready to be rendered so as to render the

previously downloaded advertisement during the next interstitial interval to the user.

*b1  
Recited*  
Claim 10 (previously amended): The media recited in claim 9 wherein the corresponding substitute methods cause the Ad Controller and Transition Sensor applets to persistently remain in the browser storage as the browser transitions across successive web pages and different web sites.

---

~~Claim 11 (canceled).~~

---

*b2*  
Claim 12 (previously added): The media recited in claim 10 wherein as a result of executing the tag, the computer determines, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and Ad Controller applets, respectively, then residing in the browser storage; and

if said new version exists on the distribution server, downloading the new version from the distribution server into the browser storage and executing the new version in lieu of the corresponding version.

Claim 13 (previously amended): The media recited in claim 10 wherein the advertising tag further comprises first and second components, the first and second components specifying the script and the advertising server, respectively.

Claim 14 (previously added): The media recited in claim 13 wherein, in response to the second component contained in the tag, the Ad Controller applet downloads the Ad

Descriptor file originating from the advertising server specified in the second component.

Claim 15 (previously added): The media recited in claim 14 wherein the Ad Descriptor file comprises a manifest of names of a plurality of predefined advertising files and associated configuration information necessary to properly play the downloaded advertisement through the browser.

Claim 16 (previously amended): The media recited in claim 15 wherein the advertising files comprise at least one media file or at least one player file necessary to render the one media file.

Claim 17 (previously amended): The media recited in claim 16 wherein the Ad Descriptor file comprises a list having: a name of each player and media file that constitutes the downloaded advertisement, a corresponding network address at which said each file can be accessed, configuration information for at least one of the player files for properly configuring a corresponding player to render an associated media file.

Claim 18 (previously added): The media recited in claim 17 wherein the Ad Controller applet comprises a play queue, and wherein the Ad Controller applet:

once all the advertising files specified in an associated Ad Descriptor file for a corresponding advertisement reside in the browser storage on the computer, inserts the associated Ad Descriptor file into an end of the play queue; and

in response to the user-initiated event and during the ensuing interstitial interval, processes advertising files

b2  
canta

specified in a specific Ad Descriptor file then situated at a head of the play queue so as to render, through the output device, an advertisement, corresponding to the specific Ad Descriptor file, to the user.

---

~~Claim 19 (canceled).~~

---

b3

Claim 20 (previously amended): The media recited in claim 18 wherein as a result of executing the tag, the computer determines, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and Ad Controller applets, respectively, then residing in the browser storage; and

if said new version exists on the distribution server, downloading the new version from the distribution server into the browser storage and executing the new version in lieu of the corresponding version.

Claim 21 (previously added): The media recited in claim 4 wherein the user-initiated event is an affirmative action taken by the user, through a browser, to navigate from the first web page to the next successive web page, wherein the action comprises a mouse click, a key depression or a user-invoked state change in a stored history of web pages previously visited by the user.

Claim 22 (previously added): The media recited in claim 21 wherein the computer executes the web browser which, in turn, executes the first web page, comprising the tag, and subsequently the agent.

Claim 23 (previously added): The media recited in claim 22 wherein the agent overrides default life cycle methods defined in the web browser with corresponding substitute methods such that the agent persistently remains in browser storage as the browser transitions across successive web pages and different web sites.

Claim 24 (previously added): The media recited in claim 23 wherein the agent comprises:

*As  
contd*  
a Transition Sensor applet; and

an Ad Controller applet;

wherein the Transition Sensor applet instantiates and starts execution of the Ad Controller applet, monitors a user click-stream so as to detect the user-initiated event, and:

instructs the Ad Controller applet to download the Ad Descriptor file for the web advertisement from the advertising server into the browser storage on the computer; and

in response to an occurrence of the event, instructs the Ad Controller applet to cease any download of a further advertisement file specified in the Ad Descriptor file, to the extent any downloading of said further advertisement file is then occurring, and to initiate processing, through the browser, of files for an advertisement that has been previously downloaded and is currently ready to be rendered so as to render the previously downloaded advertisement during the next interstitial interval to the user.

Claim 25 (previously amended): The media recited in claim 24 wherein the corresponding substitute methods cause the Ad Controller and Transition Sensor applets to persistently

b3  
could  
remain in the browser storage as the browser transitions across successive web pages and different web sites.

---

~~Claim 26 (canceled).~~

---

Claim 27 (previously added): The media recited in claim 25 wherein as a result of executing the tag, the computer determines, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and Ad Controller applets, respectively, then residing in the browser storage; and

b4  
if said new version exists on the distribution server, downloading the new version from the distribution server into the browser storage and executing the new version in lieu of the corresponding version.

Claim 28 (previously amended): The media recited in claim 25 wherein the advertising tag further comprises a component specifying the advertising server.

Claim 29 (previously added): The media recited in claim 28 wherein, in response to the second component contained in the tag, the Ad Controller applet downloads the Ad Descriptor file originating from the advertising server specified in the second component.

Claim 30 (previously added): The media recited in claim 29 wherein the Ad Descriptor file comprises a manifest of names of a plurality of predefined advertising files and associated configuration information necessary to properly play the downloaded advertisement through the browser.

Claim 31 (previously added): The media recited in claim 30 wherein the advertising files comprise at least one media file or at least one player file necessary to render an associated media file.

b4  
canceled

Claim 32 (previously amended): The media recited in claim 31 wherein the Ad Descriptor file comprises a list having: a name of each player and media file that constitutes the downloaded advertisement, a corresponding network address at which said each file can be accessed, configuration information for at least one of the player files for properly configuring a corresponding player to render an associated media file.

Claim 33 (previously added): The media recited in claim 32 wherein the Ad Controller applet comprises a play queue, and wherein the Ad Controller applet:

once all the advertising files specified in an associated Ad Descriptor file for a corresponding advertisement, reside in the browser storage on the computer, inserts the associated Ad Descriptor file into an end of the play queue; and

in response to the user-initiated event and during the ensuing interstitial interval, processes advertising files specified in a specific Ad Descriptor file then situated at a head of the play queue so as to render, through the output device, an advertisement, corresponding to the specific Ad Descriptor file, to the user.

---

~~Claim 34 (canceled).~~



Claim 35 (previously amended): The media recited in claim 33 wherein as a result of executing the tag, the computer determines, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and Ad Controller applets, respectively, then residing in the browser storage; and

b5  
if said new version exists on the distribution server, downloading the new version from the distribution server into the browser storage and executing the new version in lieu of the corresponding version.

---

~~Claim 36 (canceled).~~

---

Claim 37 (previously amended): The method recited in claim 107 wherein the code comprises an advertising tag, the network server comprises an advertising server and the information object comprises a web advertisement.

b6  
Claim 38 (previously added): The method recited in claim 37 wherein said one file comprises an Ad Descriptor file or at least one advertising file specified in the Ad Descriptor file, the advertising file being either a media file or a player file.

Claim 39 (previously added): The method recited in claim 38, wherein the advertising code comprises an advertising tag, further comprising the steps executed by the processor, in response to execution of the tag, of:

dynamically writing a plurality of predefined applet tags that collectively implement a script into the first web page; and

downloading, in response to subsequent execution of the script, an agent from a predefined distribution server into the memory and thereafter instantiating and executing the agent.

bb could  
Claim 40 (previously added): The method recited in claim 39 wherein the user-initiated event is an affirmative action taken by the user, through a web browser, to navigate from the first web page to the next successive web page, wherein the action comprises a mouse click, a key depression or a user-invoked state change in a stored history of web pages previously visited by the user.

Claim 41 (previously amended): The method recited in claim 40 further comprising the of step, performed by the processor in response to the stored executable instructions, of executing the first web page, including the tag, under the web browser, and subsequently the agent.

Claim 42 (previously added): The method recited in claim 41 comprising the step, performed by the processor, in response to execution of the agent, of overriding default life cycle methods defined in the web browser with corresponding substitute methods such that the agent persistently remains in browser storage as the browser transitions across successive web pages and different web sites.

Claim 43 (previously added): The method recited in claim 42 wherein the agent comprises a Transition Sensor applet, and an Ad Controller applet, further comprising the step, in the Transition Sensor, of:

instantiating and starting execution of the Ad Controller applet; and

monitoring a user click-stream so as to detect the user-initiated event, the monitoring step comprising the steps of:

instructing the Ad Controller applet to download the Ad Descriptor file for the web advertisement from the advertising server into the browser storage on the computer; and

b6  
could in response to an occurrence of the event, instructing the Ad Controller applet to cease any download of a further advertisement file specified in the Ad Descriptor file, to the extent any downloading of said further advertisement file is then occurring, and initiating processing, through the browser, of files for an advertisement that has been previously downloaded and is currently ready to be rendered so as to render the previously downloaded advertisement during the next interstitial interval to the user.

Claim 44 (previously amended): The method recited in claim 43 further comprising the step, as a result of the corresponding substitute methods, of causing the Ad Controller and Transition Sensor applets to persistently remain in the browser storage as the browser transitions across successive web pages and different web sites.

---

~~Claim 45 (canceled).~~

---

b7 Claim 46 (previously added): The method recited in claim 44 further comprising the steps, performed by the processor in response to executing the tag, of:

determining, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a

corresponding version, if any, of the Transition Sensor and Ad Controller applets, respectively, then residing in the browser storage; and

if said new version exists on the distribution server, downloading the new version from the distribution server into the browser storage and executing the new version in lieu of the corresponding version.

Claim 47 (previously amended): The method recited in claim 44 wherein the advertising tag further comprises first and second components, the first and second components specifying the script and the advertising server, respectively.

b7  
could  
Claim 48 (previously added): The method recited in claim 47 further comprising the step, performed by the Ad Controller applet in response to the second component contained in the tag, of downloading the Ad Descriptor file originating from the advertising server specified in the second component.

Claim 49 (previously added): The method recited in claim 48 wherein the Ad Descriptor file comprises a manifest of names of a plurality of predefined advertising files and associated configuration information necessary to properly play the downloaded advertisement through the browser.

Claim 50 (previously added): The method recited in claim 49 wherein the advertising files comprise at least one media file, or said one media file and at least one player file necessary to render the media file.

Claim 51 (previously amended): The method recited in claim 50 wherein the Ad Descriptor file comprises a list

having: a name of each player and media file that constitutes the downloaded advertisement, a corresponding network address at which said each file can be accessed, configuration information for at least one of the player files for properly configuring a corresponding player to render an associated media file.

Claim 52 (previously added): The method recited in claim 51 wherein the Ad Controller applet comprises a play queue, further comprising the steps, performed by the Ad Controller applet, of:

*b7  
contd*  
once all the advertising files specified in an associated Ad Descriptor file for a corresponding advertisement reside in the browser storage on the computer, inserting the associated Ad Descriptor file into an end of the play queue; and

in response to the user-initiated event and during the ensuing interstitial interval, processing advertising files specified in a specific Ad Descriptor file then situated at a head of the play queue so as to render, through the output device, an advertisement, corresponding to the specific Ad Descriptor file, to the user.

---

~~Claim 53 (canceled).~~

---

*b8*  
Claim 54 (previously amended): The method recited in claim 52 further comprising the steps, performed in the processor in response to executing the tag, of:

determining, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and

Ad Controller applets, respectively, then residing in the browser storage; and

if said new version exists on the distribution server, downloading the new version from the distribution server into the browser storage and executing the new version in lieu of the corresponding version.

*gg  
could*  
Claim 55 (previously added): The method recited in claim 38 wherein the user-initiated event is an affirmative action taken by the user, through a browser, to navigate from the first web page to the next successive web page, wherein the action comprises a mouse click, a key depression or a user-invoked state change in a stored history of web pages previously visited by the user.

Claim 56 (previously amended): The method recited in claim 55 further comprising the steps, performed by the processor in response to the stored executable instructions, of executing the first web page, comprising the tag, under the web browser, and subsequently the agent.

Claim 57 (previously added): The method recited in claim 56 comprising the step, performed by the processor, in response to execution of the agent of, overriding default life cycle methods defined in the web browser with corresponding substitute methods such that the agent persistently remains in browser storage as the browser transitions across successive web pages and different web sites.

Claim 58 (previously added): The method recited in claim 57 wherein the agent comprises a Transition Sensor applet, and an Ad Controller applet, further comprising the step, in the Transition Sensor, of:

instantiating and starting execution of the Ad Controller applet; and

monitoring a user click-stream so as to detect the user-initiated event, the monitoring step comprising the steps of:

instructing the Ad Controller applet to download the Ad Descriptor file for the web advertisement from the advertising server into the browser storage on the computer; and

*b9  
could* in response to an occurrence of the event, instructing the Ad Controller applet to cease any download of a further advertisement file specified in the Ad Descriptor file, to the extent any downloading of said further advertisement file is then occurring, and initiating processing, through the browser, of files for an advertisement that has been previously downloaded and is currently ready to be rendered so as to render the previously downloaded advertisement during the next interstitial interval to the user.

Claim 59 (previously amended): The method recited in claim 58 further comprising the step, as a result of the corresponding substitute methods, of causing the Ad Controller and Transition Sensor applets to persistently remain in the browser storage as the browser transitions across successive web pages and different web sites.

---

Claim 60 (canceled).

---

*b9.* Claim 61 (previously added): The method recited in claim 59 further comprising the steps, performed by the processing in response to executing the tag, of:

determining, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and Ad Controller applets, respectively, then residing in the browser storage; and

if said new version exists on the distribution server, downloading the new version from the distribution server into the browser storage and executing the new version in lieu of the corresponding version.

*by  
contd*  
Claim 62 (previously amended): The method recited in claim 59 wherein the advertising tag further comprises a component specifying the advertising server.

Claim 63 (previously added): The method recited in claim 62 further comprising the step, performed by the Ad Controller applet in response to the second component contained in the tag, of downloading the Ad Descriptor file originating from the advertising server specified in the second component.

Claim 64 (previously added): The method recited in claim 63 wherein the Ad Descriptor file comprises a manifest of names of a plurality of predefined advertising files and associated configuration information necessary to properly play the downloaded advertisement through the browser.

Claim 65 (previously amended): The method recited in claim 64 wherein the advertising files comprise at least one media file, or said one media file and at least one player file necessary to render the one media file.



Claim 66 (previously amended): The method recited in claim 65 wherein the Ad Descriptor file comprises a list having: a name of each player and media file that constitutes the downloaded advertisement, a corresponding network address at which said each file can be accessed, configuration information for at least one of the player files for properly configuring a corresponding player to render an associated media file.

*b9  
cancel*  
Claim 67 (previously added): The method recited in claim 66 wherein the Ad Controller applet comprises a play queue, further comprising the steps of:

once all the advertising files specified in an associated Ad Descriptor file for a corresponding advertisement, reside in the browser storage on the computer, inserting the associated Ad Descriptor file into an end of the play queue; and

in response to the user-initiated event and during the ensuing interstitial interval, processing advertising files specified in a specific Ad Descriptor file then situated at a head of the play queue so as to render, through the output device, an advertisement, corresponding to the specific Ad Descriptor file, to the user.

---

~~Claim 68 (canceled).~~

---

*p10*  
Claim 69 (previously amended): The method recited in claim 67 further comprising the steps, performed by the processor, in response to executing the tag, of:

determining, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and

Ad Controller applets, respectively, then residing in the browser storage; and

*b10  
canceled*  
if said new version exists on the distribution server, downloading the new version from the distribution server into the browser storage and executing the new version in lieu of the corresponding version.

---

~~Claim 70 (canceled).~~

---

Claim 71 (previously amended): The apparatus in claim 108 wherein the code comprises an advertising tag, the network server comprises an advertising server and the information object comprises a web advertisement.

*b11*  
Claim 72 (previously added): The apparatus recited in claim 71 wherein said one file comprises an Ad Descriptor file or at least one advertising file specified in the Ad Descriptor file, the advertising file being either a media file or a player file.

Claim 73 (previously added): The apparatus recited in claim 72 wherein the advertising code comprises an advertising tag and the processor, in response to execution of the tag:

dynamically writes a plurality of predefined applet tags that collectively implement a script into the first web page; and

downloads, in response to subsequent execution of the script, an agent from a predefined distribution server into the memory and thereafter instantiates and executes the agent.

Claim 74 (previously added): The apparatus recited in claim 73 wherein the user-initiated event is an affirmative action taken by the user, through a web browser, to navigate from the first web page to the next successive web page, wherein the action comprises a mouse click, a key depression or a user-invoked state change in a stored history of web pages previously visited by the user.

Claim 75 (previously amended): The apparatus recited in claim 74 wherein the processor, in response to the stored executable instructions, executes the first web page, including the tag, under the web browser, and subsequently the agent.

*b11  
cant d*  
Claim 76 (previously added): The apparatus recited in claim 75 wherein the processor, in response to execution of the agent, overrides default life cycle methods defined in the web browser with corresponding substitute methods such that the agent persistently remains in browser storage as the browser transitions across successive web pages and different web sites.

Claim 77 (previously added): The apparatus recited in claim 76 wherein the agent comprises a Transition Sensor applet and an Ad Controller applet, and the processor, during execution of the Transition Sensor:

instantiates and starts execution of the Ad Controller applet; and

monitors a user click-stream so as to detect the user-initiated event such that the processor:

instructs the Ad Controller applet to download the Ad Descriptor file for the web advertisement from the

advertising server into the browser storage on the computer;  
and

in response to an occurrence of the event,  
instructs the Ad Controller applet to cease any download of  
a further advertisement file specified in the Ad Descriptor  
file, to the extent any downloading of said further  
advertisement file is then occurring, and initiates  
processing through the browser, of files for an  
advertisement that has been previously downloaded and is  
currently ready to be rendered so as to render the  
previously downloaded advertisement during the next  
interstitial interval to the user.

*Bill  
Couta*

Claim 78 (previously amended): The apparatus recited in  
claim 77 wherein the processor, as a result of the  
corresponding substitute methods, causes the Ad Controller  
and Transition Sensor applets to persistently remain in the  
browser storage as the browser transitions across successive  
web pages and different web sites.

---

~~Claim 79 (canceled).~~

---

Claim 80 (previously added): The apparatus recited in  
claim 78 wherein the processor in response to executing the  
tag:

*11/2*

determines, through the agent, whether a new version of  
either the Transition Sensor applet or the Ad Controller  
applet then resides on the distribution server relative to a  
corresponding version, if any, of the Transition Sensor and  
Ad Controller applets, respectively, then residing in the  
browser storage; and

if said new version exists on the distribution server,  
downloads the new version from the distribution server into

the browser storage and executes the new version in lieu of the corresponding version.

Claim 81 (previously amended): The apparatus recited in claim 78 wherein the advertising tag further comprises first and second components, the first and second components specifying the script and the advertising server, respectively.

Claim 82 (previously added): The apparatus recited in claim 81 wherein the processor, during execution of the Ad Controller applet and in response to the second component contained in the tag, downloads the Ad Descriptor file originating from the advertising server specified in the second component.

Claim 83 (previously added): The apparatus recited in claim 82 wherein the Ad Descriptor file comprises a manifest of names of a plurality of predefined advertising files and associated configuration information necessary to properly play the downloaded advertisement through the browser.

Claim 84 (previously amended): The apparatus recited in claim 83 wherein the advertising files comprise at least one media file, or at least one player file necessary to render the one media file.

Claim 85 (previously amended): The apparatus recited in claim 84 wherein the Ad Descriptor file comprises a list having: a name of each player and media file that constitutes the downloaded advertisement, a corresponding network address at which said each file can be accessed, configuration information for at least one of the player

files for properly configuring a corresponding player to render an associated media file.

Claim 86 (previously added): The apparatus recited in claim 85 wherein the Ad Controller applet comprises a play queue, wherein, the processor during execution of the Ad Controller applet:

once all the advertising files specified in an associated Ad Descriptor file for a corresponding advertisement, reside in the browser storage on the computer, inserts the associated Ad Descriptor file into an end of the play queue; and

in response to the user-initiated event and during the ensuing interstitial interval, processes advertising files specified in a specific Ad Descriptor file then situated at a head of the play queue so as to render, through the output device, an advertisement, corresponding to the specific Ad Descriptor file, to the user.

*6/12  
Cantor*  
~~Claim 87 (canceled).~~

Claim 88 (previously amended): The apparatus recited in claim 86 wherein the processor in response to executing the tag:

determines, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and Ad Controller applets, respectively, then residing in the browser storage; and

if said new version exists on the distribution server, downloads the new version from the distribution server into

the browser storage and executes the new version in lieu of the corresponding version.

Claim 89 (previously added): The apparatus recited in claim 72 wherein the user-initiated event is an affirmative action taken by the user, through a browser, to navigate from the first web page to the next successive web page, wherein the action comprises a mouse click, a key depression or a user-invoked state change in a stored history of web pages previously visited by the user.

Claim 90 (previously amended): The apparatus recited in claim 89 wherein the processor in response to the stored executable instructions, executes the first web page, comprising the tag, under the web browser, and subsequently the agent.

Claim 91 (previously added): The apparatus recited in claim 90 wherein the processor, in response to execution of the agent of, overrides default life cycle methods defined in the web browser with corresponding substitute methods such that the agent persistently remains in browser storage as the browser transitions across successive web pages and different web sites.

Claim 92 (previously added): The apparatus recited in claim 91 wherein the agent comprises a Transition Sensor applet and an Ad Controller applet, and the processor, during execution of the Transition Sensor applet:

instantiates and starts execution of the Ad Controller applet; and

monitors a user click-stream so as to detect the user-initiated event such that the processor:

instructs the Ad Controller applet to download the Ad Descriptor file for the web advertisement from the advertising server into the browser storage on the computer; and

in response to an occurrence of the event, instructs the Ad Controller applet to cease any download of a further advertisement file specified in the Ad Descriptor file, to the extent any downloading of said further advertisement file is then occurring, and initiates processing, through the browser, of files for an advertisement that has been previously downloaded and is currently ready to be rendered so as to render the previously downloaded advertisement during the next interstitial interval to the user.

112  
Claim 93 (previously amended): The apparatus recited in claim 92 wherein the processor, as a result of the corresponding substitute methods, causes the Ad Controller and Transition Sensor applets to persistently remain in the browser storage as the browser transitions across successive web pages and different web sites.

Claim 94 (canceled).

Claim 95 (previously added): The apparatus recited in claim 93 wherein the processing in response to executing the tag:

determines, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and Ad Controller applets, respectively, then residing in the browser storage; and



if said new version exists on the distribution server, downloads the new version from the distribution server into the browser storage and executing the new version in lieu of the corresponding version.

Claim 96 (previously added): The apparatus recited in claim 93 wherein the output device is a display.

Claim 97 (previously amended): The apparatus recited in claim 93 wherein the advertising tag further comprises a component specifying the advertising server.

Claim 98 (previously added): The apparatus recited in claim 97 wherein the processor, during execution of the Ad Controller applet and in response to the second component contained in the tag, downloads the Ad Descriptor file originating from the advertising server specified in the second component.

Claim 99 (previously added): The apparatus recited in claim 98 wherein the Ad Descriptor file comprises a manifest of names of a plurality of predefined advertising files and associated configuration information necessary to properly play the downloaded advertisement through the browser.

Claim 100 (previously amended): The apparatus recited in claim 99 wherein the advertising files comprise at least one media file or at least one player file necessary to render the one media file.

Claim 101 (previously added): The apparatus recited in claim 100 wherein the Ad Descriptor file comprises a list having: a name of each player and media file that

constitutes the downloaded advertisement, a corresponding network address at which said each file can be accessed, configuration information for at least one of the player files for properly configuring the corresponding player to render an associated media file.

Claim 102 (previously added): The apparatus recited in claim 101 wherein the Ad Controller applet comprises a play queue, wherein the processor, in response to the stored executable instructions:

once all the advertising files specified in an associated Ad Descriptor file for a corresponding advertisement, reside in the browser storage on the computer, inserts the associated Ad Descriptor file into an end of the play queue; and

in response to the user-initiated event and during the ensuing interstitial interval, processes advertising files specified in a specific Ad Descriptor file then situated at a head of the play queue so as to render, through the output device, an advertisement, corresponding to the specific Ad Descriptor file, to the user.

Claim 104 (previously amended): The apparatus recited in claim 102 wherein the processor, in response to executing the tag:

determines, through the agent, whether a new version of either the Transition Sensor applet or the Ad Controller applet then resides on the distribution server relative to a corresponding version, if any, of the Transition Sensor and Ad Controller applets, respectively, then residing in the browser storage; and

if said new version exists on the distribution server, downloads the new version from the distribution server into

the browser storage and executes the new version in lieu of the corresponding version.

Claim 105 (previously amended): The apparatus recited in claim 102 wherein the output device is a display.

106. (currently amended; re-presented - formerly independent claim 1): A computer readable medium storing a first web page wherein the first web page comprises a plurality of computer readable instructions, the instructions representing page content and containing embedded code, wherein the code, when executed by a client computer during processing the instructions on the web page, causes the computer to:

communicate a request to a management server;  
as a result of the request, download, from a network server or the management server and while the computer renders the first web page to a user through an output device operative in conjunction with the computer, at least one file which is to be subsequently employed, by the processor, to render an information object, the information object being selected by the management server; and

in response to a user-initiated event, detected by the computer, for transitioning from the first web page to a next successive web page and which signifies a start of a next interstitial interval, suspend further downloading of files and process the one file so as to render the information object through the output device to the user during the interval; and

wherein neither the code nor the first web page references the code does not contain any reference to the information object, specifies a location of the object itself or contains any content from the object such that use

of the code eliminates a need to store either content for the information object or an address of the object within the first web page thereby substantially decoupling the object ~~content~~ from the first web page.

107. (currently amended; re-presented - formerly independent claim 36): A method for use in a computer having a processor and a memory, the memory connected to the processor and storing both computer executable instructions and a first web page, the first web page having a plurality of computer readable instructions representing page content and containing embedded code, the method comprising the steps performed by the processor, in response to the executable instructions and as a result of executing the code during processing the instructions on the web page, of:

communicating a request to a management server;

as a result of the request, downloading, from a network server or the management server and while the computer renders the first web page to a user through an output device operative in conjunction with the computer, at least one file which is to be subsequently employed, by the processor, to render an information object, the information object being selected by the management server; and

in response to a user-initiated event detected by the computer for transitioning from the first web page to a next successive web page and which signifies a start of a next interstitial interval, suspending further downloading of files and processing the one file so as to render the object through the output device to the user during the interval;

wherein neither the code nor the first web page references ~~the code does not contain any reference to the~~ information object, specifies a location of the object itself or contains any content from the object such that use

big  
could

of the code eliminates a need to store either content for the object or an address of the object within the first web page thereby substantially decoupling the object ~~content~~ from the first web page.

108. (currently amended; re-presented - formerly independent claim 70) Apparatus for rendering an information object in response to a first web page containing embedded code, the apparatus comprising:

a processor; and

a memory, the memory connected to the processor and storing both computer executable instructions and the first web page, the first web page having a plurality of computer readable instructions representing page content and the embedded code;

wherein the processor, in response to the executable instructions and as a result of executing the code during processing the instructions on the web page:

communicates a request to a management server;

as a result of the request, downloads, from a network server or the management server and while the computer renders the first web page to a user through an output device operative in conjunction with the computer, at least one file which is to be subsequently employed, by the processor, to render an information object, the information object being selected by the management server; and

in response to a user-initiated event detected by the computer for transitioning from the first web page to a next successive web page and which signifies a start of a next interstitial interval, suspends further downloading of files and processes the one file so as to render the information object through the output device to the user during the interval; and

H12  
outd

wherein neither the code nor the first web page  
references the code does not contain any reference to the  
information object, specifies a location of the object  
itself or contains any content from the object such that use  
of the code eliminates a need to store either content for  
the information object or an address of the object within  
the first web page thereby substantially decoupling the  
object ~~content~~ from the first web page.

---